

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ**

Ордена Трудового Красного Знамени

**Федеральное государственное бюджетное образовательное
учреждение высшего образования**

«Московский технический университет связи и информатики»

Лабораторная работа № 3

«Создание хэш-таблиц»

Выполнила: Студентка группы БВТ2306

Максимова Мария

Москва 2024

Гитхаб: <https://github.com/Mashaaaaa7/itip-3>

Задачи:

Задание 1:

1. Создайте класс HashTable, который будет реализовывать хэш-таблицу с помощью метода цепочек.
2. Реализуйте методы put(key, value), get(key) и remove(key), которые добавляют, получают и удаляют пары «ключ-значение» соответственно.
3. Добавьте методы size() и isEmpty(), которые возвращают количество элементов в таблице и проверяют, пуста ли она

Задание 2: Работа с встроенным классом HashMap.

Вариант 4: Реализация хэш-таблицы для хранения информации о книгах в библиотеке. Ключом будет ISBN книги, а значением - объект класса Book, содержащий информацию о названии, авторе и количестве копий. Необходимо реализовать операции вставки, поиска и удаления книги по ISBN.

При выполнении задания 1 реализуем методы put, get, remove (key).

Код к заданию 1:

```
import java.util.HashMap;
import java.util.Map;

class KeyValue {
    private String key;
    private String value;

    public KeyValue(String key, String value) {
        this.key = key;
        this.value = value;
    }

    public String getKey() {
        return key;
    }

    public String getValue() {
        return value;
    }
}

public class HashTable {
    private HashMap<String, KeyValue> keyMap;
```

```

public HashTable() {
    keyMap = new HashMap<>();
}

public void put(String key, KeyValue value) {
    keyMap.put(key, value);
}

public KeyValue get(String key) {
    return keyMap.get(key);
}

public void remove(String key) {
    keyMap.remove(key);
}

public int size() {
    return keyMap.size();
}

public boolean isEmpty() {
    return keyMap.isEmpty();
}

public static void main(String[] args) {
    HashTable hashTable = new HashTable();
    hashTable.put("name", new KeyValue("name", "John Doe"));
    hashTable.put("age", new KeyValue("age", "30"));

    System.out.println("Размер таблицы: " + hashTable.size()); // Вывод:
2
    KeyValue value = hashTable.get("name");
    if (value != null) {
        System.out.println("Значение для ключа 'name': " +
value.getValue()); // Вывод: John Doe
    } else {
        System.out.println("Ключ 'name' не найден.");
    }

    hashTable.remove("age");

    System.out.println("Размер таблицы: " + hashTable.size()); // Вывод:
1
    System.out.println("Пуста ли таблица: " + hashTable.isEmpty()); //
Вывод: false
    }
}

```

Этот код представляет простую имитацию хэш-таблицы с использованием класса HashTable.

- Класс KeyValue представляет пару ключ-значение.

- Класс `HashTable` содержит `HashMap`, в котором ключом является строка, а значением - объект класса `KeyValue`.
- Метод `put` добавляет пару ключ-значение в таблицу.
- Метод `get` возвращает значение по заданному ключу.
- Метод `remove` удаляет значение по ключу.
- Метод `size` возвращает размер таблицы (количество элементов).
- Метод `isEmpty` проверяет, пуста ли таблица.

В методе `main` создается объект `HashTable`, добавляются пары ключ-значение, выводится размер таблицы, извлекается значение по ключу `"name"`, удаляется значение по ключу `"age"` и выводится результат проверки на пустоту таблицы.

Итак, после выполнения данного кода будет выведено следующее:

- Размер таблицы: 2
- Значение для ключа `'name'`: John Doe
- Размер таблицы: 1
- Пуста ли таблица: false

Далее , перейдем к заданию 2.

Мой код:

```
import java.util.HashMap;

class Book {
    private String title;
    private String author;
    private int copies;

    public Book(String title, String author, int copies) {
        this.title = title;
        this.author = author;
        this.copies = copies;
    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
```

```

        return author;
    }

    public int getCopies() {
        return copies;
    }

    // Метод для изменения количества копий
    public void setCopies(int copies) {
        this.copies = copies;
    }
}

public class Library {
    private HashMap<String, Book> bookMap; // Явное объявление HashMap

    public Library() {
        bookMap = new HashMap<>();
    }

    public void addBook(String isbn, Book book) {
        bookMap.put(isbn, book); // Добавление книги в хэш-таблицу
    }

    public Book findBook(String isbn) {
        return bookMap.get(isbn); // Поиск книги по ISBN
    }

    public void removeBook(String isbn) {
        bookMap.remove(isbn); // Удаление книги по ISBN
    }

    public static void main(String[] args) {
        Library library = new Library();

        // Добавление книг
        library.addBook("978-0143034231", new Book("1984", "George Orwell",
5));
        library.addBook("978-0141439501", new Book("Pride and Prejudice",
"Jane Austen", 3));

        // Поиск книги
        Book foundBook = library.findBook("978-0143034231");
        if (foundBook != null) {
            System.out.println("Найдена книга: " + foundBook.getTitle() + "
by " + foundBook.getAuthor());
        } else {
            System.out.println("Книга не найдена.");
        }

        // Удаление книги
        library.removeBook("978-0141439501");

        // Поиск удаленной книги
        foundBook = library.findBook("978-0141439501");
        if (foundBook != null) {
            System.out.println("Книга найдена.");
        } else {
            System.out.println("Книга не найдена.");
        }
    }
}

```

Этот код представляет собой пример использования HashMap в Java для хранения книг в библиотеке.

Класс Book представляет книгу с полями заголовка, автора и количество копий. В классе Library создается HashMap bookMap, где ключом является ISBN книги, а значением объект книги.

Метод addBook добавляет книгу в HashMap, используя ISBN в качестве ключа. Метод findBook ищет книгу по ISBN и возвращает её. Метод removeBook удаляет книгу из HashMap по ISBN.

В методе main создается объект класса Library и добавляются две книги. Затем находится книга по ISBN, выводится информация о ней. Книга удаляется по ISBN и затем снова производится поиск этой книги, результат которого выводится в консоль.